Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Testing the Simultaneous Embeddability of Two Graphs whose Intersection is a Biconnected Graph or a Tree

Patrizio Angelini[1], Giuseppe Di Battista[1], Fabrizio Frati[1], Maurizio Patrignani[1], and Ignaz Rutter[2]

RT-DIA-175-2010 **August 2010**

[1] Dipartimento di Informatica e Automazione,
Università Roma Tre, Rome, Italy.
{angelini,gdb,frati,patrigna}@dia.uniroma3.it

[2] Institute of Theoretical Informatics
Karlsruhe Institute of Technology (KIT), Germany.
rutter@kit.edu

Supported in part by MIUR (Italy), Projects AlgoDEEP no. 2008TFBWL4 and FIRB "Advanced tracking system in intermodal freight transportation", no. RBIP06BZW8.

ABSTRACT

In this paper we study the time complexity of the problem *Simultaneous Embedding with Fixed Edges* (SEFE), that takes two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ as input and asks whether a planar drawing Γ_1 of G_1 and a planar drawing Γ_2 of G_2 exist such that: (i) each vertex $v \in V$ is mapped to the same point in Γ_1 and in Γ_2; (ii) every edge $e \in E_1 \cap E_2$ is mapped to the same Jordan curve in Γ_1 and Γ_2. First, we show a polynomial-time algorithm for SEFE when the *intersection graph* of G_1 and G_2, that is the planar graph $G_{1\cap 2} = (V, E_1 \cap E_2)$, is *biconnected*. Second, we show that SEFE, when $G_{1\cap 2}$ is a *tree*, is equivalent to a suitably-defined *book embedding* problem. Based on such an equivalence and on recent results by Hong and Nagamochi, we show a linear-time algorithm for the SEFE problem when $G_{1\cap 2}$ is a *star*.

1 Introduction

Let $G_1 = (V, E_1), \ldots, G_k = (V, E_k)$ be k graphs on the same set of vertices. A *simultaneous embedding* of $G_1, \ldots, G_k$ consists of k planar drawings $\Gamma_1, \ldots, \Gamma_k$ of $G_1, \ldots, G_k$, respectively, such that any vertex $v \in V$ is mapped to the same point in every drawing Γ_i. Because of the applications to several visualization methodologies and because of the interesting related theoretical problems, constructing simultaneous graph embeddings has recently grown up as a distinguished research topic in Graph Drawing.

The two main variants of the simultaneous embedding problem are the *geometric simultaneous embedding* and the *simultaneous embedding with fixed edges*. The former requires the edges to be straight-line segments, while the latter relaxes such a constraint by just requiring the edges that are common to distinct graphs to be represented by the same Jordan curve in all the drawings. Geometric simultaneous embedding turns out to have limited usability, as geometric simultaneous embeddings do not always exist if the input graphs are three paths [3], if they are two outerplanar graphs [3], if they are two trees [13], and even if they are a tree and a path [2]. Further, testing whether two planar graphs admit a geometric simultaneous embedding is $\mathcal{NP}$-hard [8].

On the other hand, a simultaneous embedding with fixed edges (SEFE) always exists for much larger graph classes. Namely, a tree and a path always have a SEFE with few bends per edge [7]; an outerplanar graph and a path or a cycle always have a SEFE with few bends per edge [6]; a planar graph and a tree always have a SEFE [11].

The main open question about SEFE is whether testing the existence of a SEFE of two planar graphs is doable in polynomial time or not. A number of known results are related to this problem. Namely, Gassner *et al.* proved that testing whether three planar graphs admit a SEFE is $\mathcal{NP}$-hard and that SEFE is in $\mathcal{NP}$ for any number of input graphs [12]; Fowler *et al.* characterized the planar graphs that always have a SEFE with any other planar graph and proved that testing whether two outerplanar graphs admit a SEFE is in $\mathcal{P}$ [10]; Fowler *et al.* showed how to test in polynomial time whether two planar graphs admit a SEFE if one of them contains at most one cycle [9]; Jünger and Schulz characterized the graphs $G_{1\cap 2}$ that allow for a SEFE of any two planar graphs G_1 and G_2 whose intersection graph is $G_{1\cap 2}$ [16]; Angelini *et al.* showed how to test whether two planar graphs admit a SEFE if one of them has a fixed embedding [1].

In this paper, we show the following results.

In Sect. 3 we show a cubic-time algorithm for the SEFE problem when the intersection graph $G_{1\cap 2}$ of G_1 and G_2 is biconnected. Our algorithm exploits the SPQR-tree decomposition of $G_{1\cap 2}$ in order to test whether a planar embedding of $G_{1\cap 2}$ exists that allows the edges of G_1 and G_2 not in $G_{1\cap 2}$ to be drawn in such a way that no two edges of the same graph intersect.

In Sect. 4 we show that the SEFE problem, when $G_{1\cap 2}$ is a tree, is equivalent to a suitably-defined book embedding problem. Namely, we show that, for every instance G_1, G_2 of SEFE such that $G_{1\cap 2}$ is a tree, there exist a graph G', whose edges are partitioned into two sets E'_1 and E'_2, and a set of hierarchical constraints on the set of vertices of G', such that G_1 and G_2 have a SEFE if and only if G' admits a 2-page book embedding in which the edges of E'_1 are in one page, the edges of E'_2 are in another page, and the order of the vertices of G' along the spine respects the hierarchical constraints. Based on this characterization and on recent results by Hong and Nagamochi [15] concerning 2-page book embeddings with the edges assigned to the pages in the input, we prove that linear time suffices to solve the SEFE problem when $G_{1\cap 2}$ is a star.

2 Preliminaries

A *drawing* of a graph is a mapping of each vertex to a distinct point of the plane and of each edge to a simple Jordan curve connecting its endpoints. A drawing is *planar* if the curves representing its edges do not cross but, possibly, at common endpoints. A graph is *planar* if it admits a planar drawing. Two drawings of the same graph are *equivalent* if they determine the same circular ordering around each vertex. A *planar embedding* is an equivalence class of planar drawings. A planar drawing partitions the plane into topologically connected regions, called *faces*. The unbounded face is the *outer face*.

A *Simultaneous Embedding with Fixed Edges* (SEFE) of k planar graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2), \ldots, G_k = (V, E_k)$ consists of k drawings $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$ such that: (i) Γ_i is a planar drawing of G_i, for $1 \leq i \leq k$; (ii) any vertex $v \in V$ is mapped to the same point in every drawing Γ_i, for $1 \leq i \leq k$; (iii) any edge $e \in E_i \cap E_j$ is mapped to the same Jordan curve in Γ_i and in Γ_j, for $1 \leq i, j \leq k$. The problem of testing whether k graphs admit a SEFE is called the SEFE *problem*. Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, the *intersection graph* of G_1 and G_2 is the planar graph $G_{1\cap 2} = (V, E_1 \cap E_2)$; further, the *exclusive subgraph* of G_1 (resp. of G_2) is the graph $G_{1\setminus 2} = (V, E_1 \setminus E_2)$ (resp. $G_{2\setminus 1} = (V, E_2 \setminus E_1)$). The *exclusive edges* of G_1 (of G_2) are the edges in $G_{1\setminus 2}$ (resp. in $G_{2\setminus 1}$).

A *book embedding* of a graph $G = (V, E)$ consists of a total ordering $\prec$ of the vertices in V and of an assignment of the edges in E to *pages* of a book, in such a way that no two edges (a, b) and (c, d) are assigned to the same page if $a \prec c \prec b \prec d$. A *$k$-page book embedding* is a book embedding using k pages. A *constrained k-page book embedding* is a k-page book embedding in which the assignment of edges to the pages is part of the input.

A graph is *connected* if every pair of vertices is connected by a path. A graph G is *biconnected* (resp. *triconnected*) if removing any vertex (resp. any two vertices) leaves G connected. In order to handle the decomposition of a biconnected graph into its triconnected components, we use the *SPQR-trees*, a data structure introduced by Di Battista and Tamassia (see, e.g., [4, 5]). The SPQR-tree of a biconnected graph G is depicted in Fig. 1.

SPQR-Trees

A graph is *st-biconnectible* if adding edge (s, t) to it yields a biconnected graph. Let G be an st-biconnectible graph. A *separation pair* of G is a pair of vertices whose removal disconnects the graph. A *split pair* of G is either a separation pair or a pair of adjacent vertices. A *maximal split component* of G with respect to a split pair $\{u, v\}$ (or, simply, a maximal split component of $\{u, v\}$) is either an edge (u, v) or a maximal subgraph G' of G such that G' contains u and v, and $\{u, v\}$ is not a split pair of G'. A vertex $w \neq u, v$ belongs to exactly one maximal split component of $\{u, v\}$. We call *split component* of $\{u, v\}$ the union of any number of maximal split components of $\{u, v\}$.

The SPQR-tree $\mathcal{T}$ of a biconnected graph G is a tree rooted at an edge e of G, called *reference edge*, describing the recursive decomposition of G induced by its split pairs. The

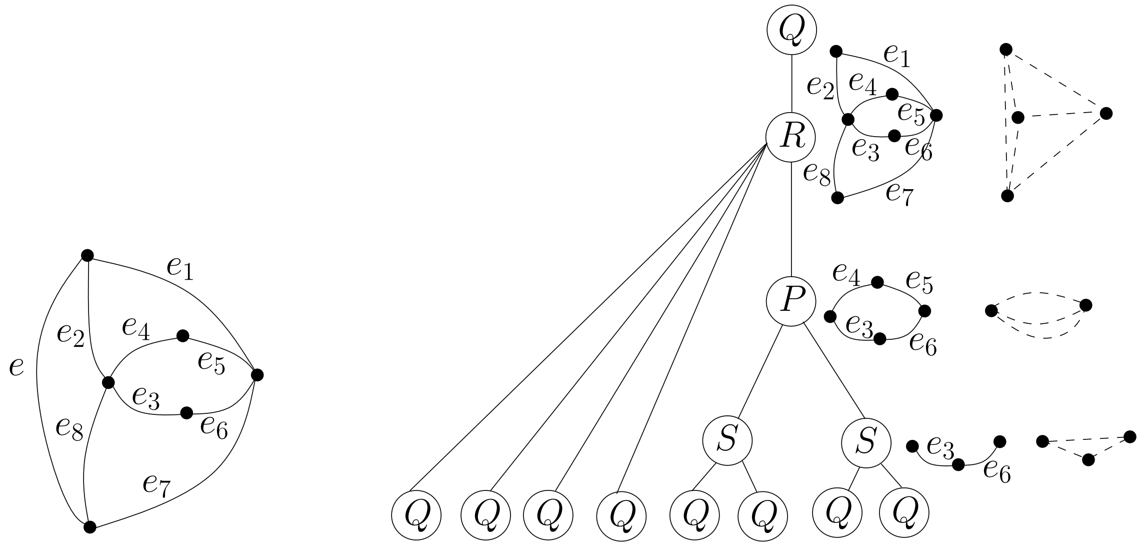

Figure 1: A biconnected planar graph G and its SPQR-tree $\mathcal{T}$. The pertinent graphs and the skeletons of an R-node, of a P-node, and of an S-node are shown.

nodes of $\mathcal{T}$ are of four types: S, P, Q, and R.

Each node μ of $\mathcal{T}$ has an associated st-biconnectible multigraph, called the *skeleton* of μ and denoted by $skel(\mu)$. Skeleton $skel(\mu)$ shows how the children of μ, represented by "virtual edges", are arranged into μ. The virtual edge in $skel(\mu)$ associated with a child node ν, is called the *virtual edge of ν in $skel(\mu)$*. For each virtual edge e_i of $skel(\mu)$, recursively replace e_i with the skeleton $skel(\mu_i)$ of its corresponding child μ_i. The subgraph of G that is obtained in this way is the *pertinent graph* of μ and is denoted by $G(\mu)$. The skeleton of a node μ also contains a virtual edge representing the *rest of the graph*, that is, the graph obtained from G by removing all the vertices of $G(\mu)$, except for its poles, together with their incident edges. We say that a vertex v of G *belongs* to a node μ of $\mathcal{T}$ if v is a vertex of $G(\mu)$. In this case we also say that μ *contains* v.

Given a biconnected graph G and a reference edge $e = (u', v')$, tree $\mathcal{T}$ is recursively defined as follows. At each step, a split component G^*, a pair of vertices $\{u, v\}$, and a node ν in $\mathcal{T}$ are given. A node μ corresponding to G^* is introduced in $\mathcal{T}$ and attached to its parent ν. Vertices u and v are the *poles* of μ and denoted by $u(\mu)$ and $v(\mu)$, respectively. The decomposition possibly recurs on some split components of G^*. At the beginning of the decomposition $G^* = G - \{e\}$, $\{u, v\} = \{u', v'\}$, and ν is a Q-node corresponding to e.

Base Case: If G^* consists of exactly one edge between u and v, then μ is a Q-node whose skeleton is G^* itself.

Parallel Case: If G^* is composed of at least two maximal split components $G_1, \dots, G_k$ ($k \geq 2$) of G with respect to $\{u, v\}$, then μ is a P-node. Graph $skel(\mu)$ consists of k parallel virtual edges between u and v, denoted by $e_1, \dots, e_k$ and corresponding to $G_1, \dots, G_k$, respectively. The decomposition recurs on $G_1, \dots, G_k$, with $\{u, v\}$ as pair of vertices for every graph, and with μ as parent node.

Series Case: If G^* is composed of exactly one maximal split component of G with respect to $\{u, v\}$ and if G^* has cutvertices $c_1, \dots, c_{k-1}$ ($k \geq 2$), appearing in this order on a path from u to v, then μ is an S-node. Graph $skel(\mu)$ is the path $e_1, \dots, e_k$, where virtual edge e_i connects c_{i-1} with c_i ($i = 2, \ldots, k-1$), e_1 connects u with c_1, and e_k connects c_{k-1} with v. The decomposition recurs on the split components corresponding to each of $e_1, e_2, \ldots, e_{k-1}, e_k$ with μ as parent node, and with $\{u, c_1\}, \{c_1, c_2\}, \ldots, \{c_{k-2}, c_{k-1}\}, \{c_{k-1}, v\}$ as pair of vertices, respectively.

Rigid Case: If none of the above cases applies, the purpose of the decomposition step is that of partitioning G^* into the minimum number of split components and recurring on each of them. We need some further definition. Given a maximal split component G' of a split pair $\{s, t\}$ of G^*, a vertex $w \in G'$ *properly belongs* to G' if $w \neq s, t$. Given a split pair $\{s, t\}$ of G^*, a maximal split component G' of $\{s, t\}$ is *internal* if neither u nor v (the poles of G^*) properly belongs to G', *external* otherwise. A *maximal split pair* $\{s, t\}$ of G^* is a split pair of G^* that is not contained into an internal maximal split component of any other split pair $\{s', t'\}$ of G^*. Let $\{u_1, v_1\}, \ldots, \{u_k, v_k\}$ be the maximal split pairs of G^* ($k \geq 1$) and, for $i = 1, \ldots, k$, let G_i be the union of all the internal maximal split components of $\{u_i, v_i\}$. Observe that each vertex of G^* either properly belongs to exactly one G_i or belongs to some maximal split pair $\{u_i, v_i\}$. Node μ is an R-node. Graph $skel(\mu)$ is the graph obtained from G^* by replacing each subgraph G_i with the virtual edge e_i between u_i and v_i. The decomposition recurs on each G_i with μ as parent node and with $\{u_i, v_i\}$ as pair of vertices.

For each node μ of $\mathcal{T}$, the construction of $skel(\mu)$ is completed by adding a virtual edge (u, v) representing the rest of the graph.

The SPQR-tree $\mathcal{T}$ of a graph G with n vertices and m edges has m Q-nodes and $O(n)$ S-, P-, and R-nodes. Also, the total number of vertices of the skeletons stored at the nodes of $\mathcal{T}$ is $O(n)$. SPQR-trees can be constructed and handled efficiently. Namely, given a biconnected planar graph G, the SPQR-tree $\mathcal{T}$ of G can be computed in linear time [4, 5, 14].

In the following, we will only refer to the SPQR-tree of the intersection graph $G_{1\cap 2}$ of two graphs G_1 and G_2. However, with a slight abuse of notation, we will denote by $G_1(\mu)$ (by $G_2(\mu)$) the subgraph of G_1 (of G_2) induced by the vertices in $G_{1\cap 2}(\mu)$.

3 The Intersection Graph is Biconnected

In this section we describe an algorithm for computing a SEFE of two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ when the intersection graph $G_{1\cap 2}$ is biconnected. Denote by $\mathcal{T}$ the SPQR-tree of $G_{1\cap 2}$.

To ease the description of the algorithm, we assume that $\mathcal{T}$ is rooted at any edge e of $G_{1\cap 2}$. Such an assumption implies that e is adjacent to the outer face of any computed embedding of $G_{1\cap 2}$. Observe that this does not preclude the possibility of finding a SEFE of G_1 and G_2. Namely, consider any SEFE in the plane; "wrap" the SEFE around a sphere; project the SEFE back to the plane from a point in a face incident to e, thus obtaining a SEFE of G_1 and G_2 in which e is incident to the outer face of the embedding of $G_{1\cap 2}$. Furthermore, if e is the parent in $\mathcal{T}$ of an S-node, subdivide the edge of $\mathcal{T}$ connecting e to its only child by inserting a P-node. Observe that the described insertion of an "artificial" P-node ensures that the parent of any S-node is either an R-node or a P-node.

For every P-node and R-node μ of $\mathcal{T}$, the *visible nodes* of μ are the children of μ that are not S-nodes plus the children of each child of μ that is an S-node.

An exclusive edge e of G_1 or of G_2 is an *internal edge* of a node $\mu \in \mathcal{T}$ if both the end-vertices of e belong to μ, at least one of them is not a pole of μ, and there exists no descendant of μ containing both the end-vertices of e. An exclusive edge e of G_1 or of G_2 is an *outer edge* of a node $\mu \in \mathcal{T}$ if exactly one end-vertex of e belongs to μ and this end-vertex is not a pole of μ. An exclusive edge e of G_1 or of G_2 is an *intra-pole edge* of a node $\mu \in \mathcal{T}$ if its end-vertices are the poles of μ. Observe that an exclusive edge e of G_1 or of G_2 can be an outer edge of a linear number of nodes of $\mathcal{T}$; further, e is an internal edge of at most one node of $\mathcal{T}$; moreover, e can be an intra-pole edge of a linear number of nodes of $\mathcal{T}$; however, e can be an intra-pole edge of at most one P-node of $\mathcal{T}$. In Fig. 2, edge e_1 is an internal edge of μ and an outer edge of $\rho_{1,2}$, of $\rho_{2,2}$, of ν_1, and of ν_2; edge e_2 is an internal edge of ν_2 and an outer edge of $\rho_{2,2}$ and $\rho_{2,4}$; edge e_3 is an internal edge of μ and an outer edge of $\rho_{1,3}$, ν_1, and ν_2; edge e_4 is an intra-pole edge of $\rho_{2,5}$; edge e_5 is an outer edge of $\rho_{1,2}$, of ν_1, and of μ.

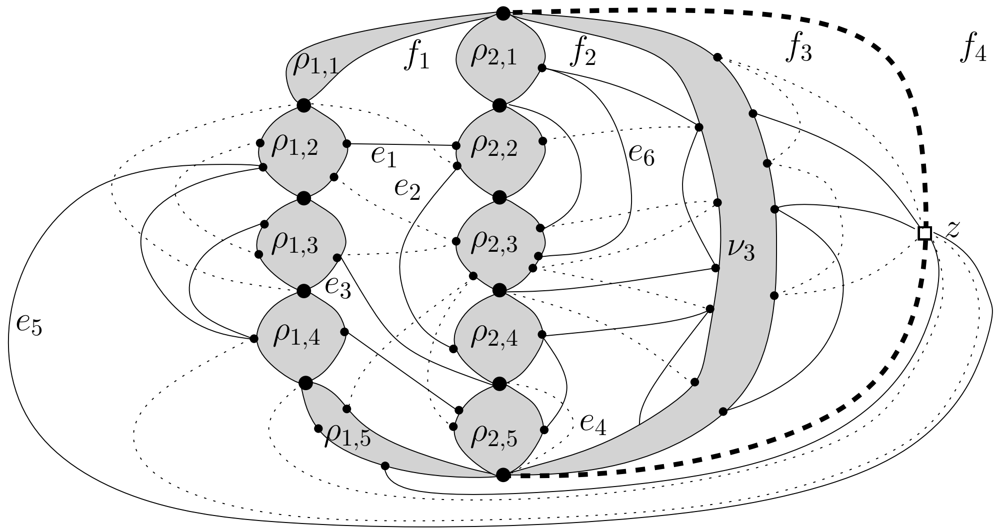

Figure 2: A SEFE of graphs $G_1(\mu)$ and $G_2(\mu)$ when μ is a P-node with three children ν_1, ν_2, and ν_3. Also, ν_1 and ν_2 have children $\rho_{1,1}, \dots, \rho_{1,5}$ and $\rho_{2,1}, \dots, \rho_{2,5}$, respectively. For each visible node τ of μ, the interior of the cycle delimiting the outer face of $G_{1\cap 2}(\tau)$ is gray. Solid (dotted) edges are exclusive edges of G_1 (G_2). The dashed edge represents the rest of the graph.

We have the following lemmata.

Lemma 1 *Let $\mathcal{E}_{1\cap 2}(\mu)$ be an embedding of $G_{1\cap 2}(\mu)$, with $\mu \in \mathcal{T}$, and let e be an internal edge of μ. Then, G_1 and G_2 have a* SEFE *in which the embedding of $G_{1\cap 2}(\mu)$ is $\mathcal{E}_{1\cap 2}(\mu)$ only if both end-vertices of e are incident to the same face of $\mathcal{E}_{1\cap 2}(\mu)$.*

Proof: Suppose, for a contradiction, that G_1 and G_2 have a SEFE in which the embedding of $G_{1\cap 2}(\mu)$ is $\mathcal{E}_{1\cap 2}(\mu)$ and the end-vertices of e are not both incident to the same face of $\mathcal{E}_{1\cap 2}(\mu)$. Then e crosses $G_{1\cap 2}(\mu)$, hence either two edges of G_1 or two edges of G_2 cross (depending on whether $e \in G_1$ or $e \in G_2$), a contradiction. □

Lemma 2 *Let $\mathcal{E}_{1\cap 2}(\mu)$ be an embedding of $G_{1\cap 2}(\mu)$, with $\mu \in \mathcal{T}$, and let e be an outer edge incident to μ in a vertex $u(e)$. Then, G_1 and G_2 have a* SEFE *in which the embedding of $G_{1\cap 2}(\mu)$ is $\mathcal{E}_{1\cap 2}(\mu)$ only if $u(e)$ is on the outer face of $\mathcal{E}_{1\cap 2}(\mu)$.*

Proof: Suppose, for a contradiction, that G_1 and G_2 have a SEFE in which the embedding of $G_{1\cap 2}(\mu)$ is $\mathcal{E}_{1\cap 2}(\mu)$ and $u(e)$ is not incident to the outer face of $\mathcal{E}_{1\cap 2}(\mu)$. Then e crosses $G_{1\cap 2}(\mu)$, hence either two edges of G_1 or two edges of G_2 cross (depending on whether $e \in G_1$ or $e \in G_2$), a contradiction. □

The algorithm performs a bottom-up traversal of $\mathcal{T}$. When the algorithm visits a node μ of $\mathcal{T}$, either it concludes that a SEFE of G_1 and G_2 does not exist, or it determines a SEFE $\Gamma(\mu)$ of $G_1(\mu)$ and $G_2(\mu)$ such that, if a SEFE of G_1 and G_2 exists, there exists one in which the SEFE of $G_1(\mu)$ and $G_2(\mu)$ is $\Gamma(\mu)$. The embedding $\Gamma(\mu)$ is computed by composing and possibly flipping the already computed embeddings of the descendants of μ. The rest of the graph, that is, the union of the graphs obtained from G_1 and G_2 by respectively removing the vertices of $G_1(\mu)$ and $G_2(\mu)$, except for $u(\mu)$ and $v(\mu)$, and their incident edges, will be placed in the same connected region of $\Gamma(\mu)$. Such a region is called the *outer face* of $\Gamma(\mu)$. The computed SEFE $\Gamma(\mu)$ of $G_1(\mu)$ and $G_2(\mu)$ is such that all the outer edges of μ can be *drawn towards the outer face*, that is, a vertex z can be inserted into the outer face of $\Gamma(\mu)$ and all the outer edges of μ can be drawn with z replacing their end-vertex not in μ, still maintaining the planarity of the drawings of $G_1(\mu)$ and $G_2(\mu)$. An example of insertion of z in a SEFE of $G_1(\mu)$ and $G_2(\mu)$ is shown in Fig. 2.

We are now ready to state the algorithm and to prove its correctness. We will later show that it has a polynomial-time implementation.

If μ is a Q-node, then $G_1(\mu)$, $G_2(\mu)$, and $G_{1\cap 2}(\mu)$ have exactly one embedding, hence no embedding choices have to be done.

If μ is an S-node, some information about the parent of μ in $\mathcal{T}$ are needed in order to decide an embedding for $G_1(\mu)$, $G_2(\mu)$, and $G_{1\cap 2}(\mu)$. Hence, such a decision is deferred to the step in which the parent of μ is analyzed.

If μ is a P-node, then, since for each visible node τ of μ the embeddings of $G_1(\tau)$ and $G_2(\tau)$ are already decided (up to a flip), an embedding of $G_1(\mu)$ and $G_2(\mu)$ is specified by an embedding of $skel(\mu)$, that is, an ordering of the nodes ν_i around the poles of μ, by a flip for each visible node τ of μ, and by an embedding of all the exclusive edges of $G_1(\mu)$ and $G_2(\mu)$ that have not yet been embedded (that is, the outer edges of μ, the internal edges of μ, the internal edges of the S-nodes children of μ, and the intra-pole edge of μ).

First, we determine an embedding $\mathcal{E}(skel(\mu))$ of $skel(\mu)$. Consider any child ν_i of μ that has an outer edge e. If e is an internal edge of μ, then e is an outer edge of a child ν_k of μ, with $k \neq i$; hence, by Lemma 1, ν_i and ν_k have to be consecutive around the poles of μ. If e is not an internal edge of μ, then e is an outer edge of μ; hence, by Lemma 2, ν_i and the virtual edge representing the rest of the graph in $skel(\mu)$ have to be consecutive around the poles of μ. Consider the graph O which has a vertex for each virtual edge of $skel(\mu)$, and which has an edge between two vertices if the corresponding virtual edges have to be consecutive around the poles of μ. If O is not a simple cycle and is not a collection of paths and isolated vertices, then we conclude that G_1 and G_2 have no SEFE. Otherwise, consider as the embedding $\mathcal{E}(skel(\mu))$ of $skel(\mu)$ any ordering of the virtual edges of $skel(\mu)$ around the poles of μ such that any two adjacent vertices in O are consecutive.

Second, we determine a flip for each visible node τ of μ and an embedding of all the exclusive edges of $G_1(\mu)$ and $G_2(\mu)$ that have not been embedded when processing the visible nodes of μ. In order to do this, we will use some auxiliary graphs. For each face f_k of $\mathcal{E}(skel(\mu))$, denote by $\nu_1(f_k)$ and by $\nu_2(f_k)$ the nodes of $\mathcal{T}$ corresponding to the two virtual edges adjacent to f_k in $skel(\mu)$ (recall that one of such virtual edges might be the one representing the rest of the graph), and construct two graphs F_k^1 and F_k^2 as follows. The nodes of F_k^1 (resp. of F_k^2) are the edges e of $G_{1\setminus 2}$ (resp. of $G_{2\setminus 1}$) such that: (i) e is an internal edge of μ, an outer edge of $\nu_1(f_k)$, and an outer edge of $\nu_2(f_k)$, or (ii) e is an internal edge of an S-node (either $\nu_1(f_k)$ or $\nu_2(f_k)$) child of μ, or (iii) e is an outer edge of μ and f_k is incident in $\mathcal{E}(skel(\mu))$ to the virtual edge of $skel(\mu)$ representing the rest of the graph and to the virtual edge containing the end-vertex of e in μ, or (iv) e is an intra-pole edge of μ. Informally speaking, the nodes of F_k^1 (resp. of F_k^2) are the edges of $G_{1\setminus 2}$ (resp. of $G_{2\setminus 1}$) that have not yet been embedded by the algorithm after processing all the visible nodes of μ and that could be embedded into f_k. The edges of F_k^1 (resp. of F_k^2) connect two vertices of F_k^1 (resp. of F_k^2) whose corresponding edges cross if they are both embedded inside f_k. In the example of Fig. 2, vertices e_2 and e_6 and edge (e_2, e_6) belong to graphs F_1^1 and F_2^1.

Denote by τ any visible node of μ. Observe that:

- Deciding the flip for $G_{1\cap 2}(\tau)$ determines the face of $\mathcal{E}(skel(\mu))$ into which the outer edges of τ have to be embedded. Namely, once a flip for $G_{1\cap 2}(\tau)$ has been fixed, there is exactly one face of $\mathcal{E}(skel(\mu))$ into which each of its outer edges can be embedded without crossing $G_{1\cap 2}(\mu)$. In the example of Fig. 2, fixing the flip for $\rho_{2,3}$ determines that e_6 is embedded into f_1 or into f_2.

- Embedding an outer edge e of τ into a face f_k determines a flip for τ. Namely, there is exactly one flip of τ that brings the end-vertex of e in τ to be incident to f_k. In the example of Fig. 2, fixing the embedding of e_2 into f_1 or into f_2 determines the flip of $\rho_{2,4}$.

- Embedding an edge e of $G_{1\setminus 2}$ (resp. of $G_{2\setminus 1}$), that is represented by a node in F_k^1 (resp. in F_k^2), into a face f_k determines that an edge e' such that (e, e') belongs to F_k^1 (resp. to F_k^2) can not be embedded into f_k. Observe that, if e' can not be embedded into f_k and e' is not an intra-pole edge of μ, then there is at most one face $f_{k'}$ with $k' \neq k$ into which e' can be embedded without crossing $G_{1\cap 2}(\mu)$. In the example of Fig. 2, fixing the embedding of e_2 into f_1 determines that e_6 is embedded into f_2.

Our algorithm uses two sets E_f and T_f. Set E_f contains the edges of $G_{1\setminus 2}$ and of $G_{2\setminus 1}$ that belong to some graph F_k^1 or F_k^2, that have been already embedded into a face of $\mathcal{E}(skel(\mu))$, and that have not yet been processed by the algorithm. Set T_f contains the visible nodes of μ whose flip has already been decided and that have not yet been processed by the algorithm. When the algorithm processes the elements of E_f and T_f, it propagates to other edges and visible nodes the embedding choices already performed on such elements.

We initialize E_f as follows. For every exclusive edge e of G_1 or of G_2 in μ that is an outer edge of two components $\nu_1(f_k)$ and $\nu_2(f_k)$, embed e into f_k and add e to E_f. Note that the embedded edges are all the internal edges and the outer edges of μ. Further, if there exists an intra-pole edge $e \in G_1$ (resp. $e \in G_2$) of μ, then embed e into any face f_k of $\mathcal{E}(skel(\mu))$ such that no edge e' has already been embedded into f_k, with $(e,e') \in F^1_k$ (resp. with $(e,e') \in F^2_k$). If no such a face f_k exists, then conclude that G_1 and G_2 have no SEFE, otherwise add e to E_f.

Next, we repeatedly apply the following procedure, called *Embedding-Flipping Step*, till the flip of every visible node of μ and the embedding of every edge represented by a node in some graph F^1_k or F^2_k have been decided, or till the algorithm returns that there is no SEFE of G_1 and G_2.

Embedding-Flipping Step

- Case 1: $E_f \neq \emptyset$. Consider any edge $e \in E_f$. Remove e from E_f.

  For each edge e' such that e and e' are adjacent in some graph F^1_k or F^2_k the following operations are performed: (i) if e has been embedded into f_k, if e' has not yet been embedded into any face of $\mathcal{E}(skel(\mu))$, and if e' belongs to a graph $F^1_{k'}$ or $F^2_{k'}$, with $k' \neq k$, then embed e' into $f_{k'}$ and add e' to E_f; (ii) if e has been embedded into f_k, if e' has not yet been embedded into any face of $\mathcal{E}(skel(\mu))$, and if e' does not belong to a graph $F^1_{k'}$ or $F^2_{k'}$, with $k' \neq k$, then conclude that G_1 and G_2 have no SEFE; (iii) if e has been embedded into a face $f_{k'} \neq f_k$ and if e' has not yet been embedded into any face of $\mathcal{E}(skel(\mu))$, then embed e' into f_k and add e' to E_f.

  If e is the outer edge of a visible node τ of μ, then: (i) if no flip has yet been decided for τ, then flip τ so that the end-vertex of e in τ is incident to the face of $\mathcal{E}(skel(\mu))$ into which e has been embedded and add τ to T_f; (ii) if a flip has already been decided for τ such that the end-vertex of e in τ is not incident to the face of $\mathcal{E}(skel(\mu))$ into which e has been embedded, then conclude that G_1 and G_2 have no SEFE.

- Case 2: $E_f = \emptyset$ and $T_f \neq \emptyset$. Consider any node $\tau \in T_f$. Remove τ from T_f.

  For each outer edge e of τ: (i) if e has not yet been embedded into any face of $\mathcal{E}(skel(\mu))$, then embed e into the face of $\mathcal{E}(skel(\mu))$ the end-vertex of e in τ is incident to and add e to E_f; (ii) if e has already been embedded into a face of $\mathcal{E}(skel(\mu))$ and the end-vertex of e in τ is not incident to such a face, then conclude that G_1 and G_2 have no SEFE.

- Case 3: $E_f = \emptyset$ and $T_f = \emptyset$. If all the edges of $G_{1\setminus 2}$ and $G_{2\setminus 1}$ that are represented by nodes in some graph F^1_k or F^2_k have been embedded into a face of $\mathcal{E}(skel(\mu))$ and if all the visible nodes of μ have been flipped, then the procedure stops. Otherwise, if there is a visible node of μ that still has to be flipped, then flip it either way and insert such a node into T_f. If there is no visible node of μ that still has to be flipped, then there is an edge e of $G_{1\setminus 2}$ and $G_{2\setminus 1}$ that is represented by a node in some graph F^1_k or F^2_k that still has to be embedded. Embed e into any face incident to both the visible nodes of μ the end-vertices of e belong to and add e to E_f.

If μ is an R-node, then the algorithm behaves exactly as in the P-node case, except that the phase in which the embedding $\mathcal{E}(skel(\mu))$ of $skel(\mu)$ is chosen is missing, as $skel(\mu)$ has exactly one planar embedding (up to a flip of the entire embedding).

We now prove the correctness of the algorithm.

Lemma 3 *A* SEFE *of G_1 and G_2 exists if and only if the described algorithm returns a* SEFE *of G_1 and G_2.*

Proof: One implication is trivial: If the algorithm returns a SEFE of G_1 and G_2, then a SEFE of G_1 and G_2 exists. We now prove the other implication.

Consider any SEFE Γ of G_1 and G_2 and, for any node μ of the SPQR-tree $\mathcal{T}$ of $G_{1\cap 2}$, consider the outer face of $G_{1\cap 2}(\mu)$ in Γ. Such a face is delimited by two paths $P^a(\Gamma,\mu)$ and $P^b(\Gamma,\mu)$ connecting $u(\mu)$ and $v(\mu)$. Consider such paths as starting at $u(\mu)$ and ending at $v(\mu)$. Denote by $L_1^a(\Gamma,\mu)$ the ordered list of vertices incident to the outer edges of μ that belong to G_1 and whose end-vertex in μ belongs to $P^a(\Gamma,\mu)$. Such vertices are ordered in $L_1^a(\Gamma,\mu)$ as they are ordered in $P^a(\Gamma,\mu)$. List $L_1^b(\Gamma,\mu)$ is defined analogously, with $P^b(\Gamma,\mu)$ replacing $P^a(\Gamma,\mu)$. Lists $L_2^a(\Gamma,\mu)$ and $L_2^b(\Gamma,\mu)$ are defined analogously to $L_1^a(\Gamma,\mu)$ and $L_1^b(\Gamma,\mu)$, with G_2 replacing G_1.

Suppose that μ is not an S-node, the following claim asserts that each of $L_1^a(\Gamma,\mu)$, $L_1^b(\Gamma,\mu)$, $L_2^a(\Gamma,\mu)$, and $L_2^b(\Gamma,\mu)$ is the same in any SEFE Γ of G_1 and G_2, that is, the structure of the outer face of $G_{1\cap 2}(\mu)$ does not depend on the choices made by the algorithm.

Claim 1 *In any* SEFE *Γ of G_1 and G_2, for any node $\mu \in \mathcal{T}$ that is not an S-node, lists $L_1^a(\Gamma,\mu)$, $L_1^b(\Gamma,\mu)$, $L_2^a(\Gamma,\mu)$, and $L_2^b(\Gamma,\mu)$ are the same, up to simultaneous swaps of $L_1^a(\Gamma,\mu)$ with $L_1^b(\Gamma,\mu)$ and of $L_2^a(\Gamma,\mu)$ with $L_2^b(\Gamma,\mu)$.*

Proof: Suppose, for a contradiction, that there exist two SEFEs and a node of $\mathcal{T}$ that is not an S-node for which the statement does not hold. We will show that this implies that one of the two SEFEs is not correct.

Consider a node $\mu \in \mathcal{T}$ that is not an S-node, for which the statement does not hold, and such that for all the descendants of μ in $\mathcal{T}$ the statement holds.

If μ is a Q-node, then $L_1^a(\Gamma,\mu)$, $L_1^b(\Gamma,\mu)$, $L_2^a(\Gamma,\mu)$, and $L_2^b(\Gamma,\mu)$ are empty lists and the statement holds, thus obtaining a contradiction.

If μ is an R-node, consider any two SEFEs Γ and Γ' of G_1 and G_2 such that not all the following four equalities hold $L_1^a(\Gamma,\mu) = L_1^a(\Gamma',\mu)$, $L_1^b(\Gamma,\mu) = L_1^b(\Gamma',\mu)$, $L_2^a(\Gamma,\mu) = L_2^a(\Gamma',\mu)$, and $L_2^b(\Gamma,\mu) = L_2^b(\Gamma',\mu)$, and such that not all the following four equalities hold $L_1^a(\Gamma,\mu) = L_1^b(\Gamma',\mu)$, $L_1^b(\Gamma,\mu) = L_1^a(\Gamma',\mu)$, $L_2^a(\Gamma,\mu) = L_2^b(\Gamma',\mu)$, and $L_2^b(\Gamma,\mu) = L_2^a(\Gamma',\mu)$. Since the statement holds for every visible node of μ and since $skel(\mu)$ has one planar embedding, up to a reversal of the adjacency lists of all the vertices, there exists a visible node of μ that is flipped differently in Γ and in Γ' and that has an outer edge e that is also an outer edge of μ. Denote by $u(e)$ the end-vertex of e in μ. Suppose that $u(e)$ is incident to the outer face of $G_{1\cap 2}(\mu)$ in Γ. Then, $u(e)$ is not incident to the outer face of $G_{1\cap 2}(\mu)$ in Γ'. It follows that edge e crosses $G_{1\cap 2}(\mu)$ in Γ', a contradiction.

If μ is a P-node, then the (at most) two children ν_x and ν_y of μ that contain end-vertices of outer edges of μ are incident to the outer face of $G_{1\cap 2}(\mu)$ in any SEFE of G_1 and G_2. The flips of ν_x and ν_y (if they are not S-nodes) or the flips of the children of ν_x and ν_y (if they are S-nodes) determine lists $L_1^a(\Gamma,\mu)$, $L_1^b(\Gamma,\mu)$, $L_2^a(\Gamma,\mu)$, and $L_2^b(\Gamma,\mu)$ in any SEFE Γ of G_1 and G_2. Then, consider any two SEFE Γ and Γ' of G_1 and G_2 such that not all the following four equalities hold $L_1^a(\Gamma,\mu) = L_1^a(\Gamma',\mu)$, $L_1^b(\Gamma,\mu) = L_1^b(\Gamma',\mu)$, $L_2^a(\Gamma,\mu) = L_2^a(\Gamma',\mu)$, and $L_2^b(\Gamma,\mu) = L_2^b(\Gamma',\mu)$, and such that not all the following four equalities hold $L_1^a(\Gamma,\mu) = L_1^b(\Gamma',\mu)$, $L_1^b(\Gamma,\mu) = L_1^a(\Gamma',\mu)$, $L_2^a(\Gamma,\mu) = L_2^b(\Gamma',\mu)$, and $L_2^b(\Gamma,\mu) = L_2^a(\Gamma',\mu)$. Similarly to the R-node case, if a visible node of μ has an outer edge e that is also an outer edge of μ and such a node is flipped differently in Γ and in Γ', then the end-vertex $u(e)$ of e in μ is not incident to the outer face of $G_{1\cap 2}(\mu)$ either in Γ or in Γ'. It follows that edge e crosses $G_{1\cap 2}(\mu)$ in Γ or in Γ', a contradiction. □

The following claim asserts that the structure of the outer face of $G_{1\cap 2}(\mu)$ (and of the exclusive edges of G_1 and G_2 embedded into it) is the only property coming from an embedding of $G_1(\mu)$ and $G_2(\mu)$ that affects the possibility of constructing a SEFE of the rest of the graph.

Claim 2 *Suppose that a* SEFE *of G_1 and G_2 exists. Then, for any node $\mu \in \mathcal{T}$ that is not an S-node, any* SEFE *of $G_1(\mu)$ and $G_2(\mu)$ in which the outer edges of μ can be drawn towards the outer face can be extended into a* SEFE *of G_1 and G_2.*

Proof: Consider any SEFE Γ of G_1 and G_2 and consider any SEFE $\Gamma(\mu)$ of $G_1(\mu)$ and $G_2(\mu)$ in which the outer edges of μ can be drawn towards the outer face.

Similarly to the proof of Claim 1, it can be proved that if neither $L_1^a(\Gamma,\mu) = L_1^a(\Gamma(\mu),\mu)$, $L_1^b(\Gamma,\mu) = L_1^b(\Gamma(\mu),\mu)$, $L_2^a(\Gamma,\mu) = L_2^a(\Gamma(\mu),\mu)$, and $L_2^b(\Gamma,\mu) = L_2^b(\Gamma(\mu),\mu)$ nor $L_1^a(\Gamma,\mu) = L_1^b(\Gamma(\mu),\mu)$, $L_1^b(\Gamma,\mu) = L_1^a(\Gamma(\mu),\mu)$, $L_2^a(\Gamma,\mu) = L_2^b(\Gamma(\mu),\mu)$, and $L_2^b(\Gamma,\mu) = L_2^a(\Gamma(\mu),\mu)$ holds, then there is an end-vertex of an outer edge of μ that either is not incident to the outer face of $G_{1\cap 2}(\mu)$ in $\Gamma(\mu)$, thus contradicting the fact that the outer edges of μ can be drawn towards the outer face in $\Gamma(\mu)$, or is not incident to the outer face of $G_{1\cap 2}(\mu)$ in Γ, thus contradicting the fact that Γ is a SEFE. By suitably flipping Γ, we can hence assume that $L_1^a(\Gamma,\mu) = L_1^a(\Gamma(\mu),\mu)$, $L_1^b(\Gamma,\mu) = L_1^b(\Gamma(\mu),\mu)$, $L_2^a(\Gamma,\mu) = L_2^a(\Gamma(\mu),\mu)$, and $L_2^b(\Gamma,\mu) = L_2^b(\Gamma(\mu),\mu)$.

Remove from Γ the drawing of $G_1(\mu)$ and $G_2(\mu)$, except for $u(\mu)$ and $v(\mu)$. Insert $\Gamma(\mu)$ inside the face of the modified Γ into which the previous drawing of $G_1(\mu)$ and $G_2(\mu)$ used to lie; the modified Γ is scaled up till the insertion of $\Gamma(\mu)$ does not cause crossings among the edges of the modified Γ and those of $\Gamma(\mu)$. Continuously deform the edges incident to $u(\mu)$ and $v(\mu)$ in Γ so that they end at the points where $u(\mu)$ and $v(\mu)$ are drawn in $\Gamma(\mu)$. This can always be done since $u(\mu)$ and $v(\mu)$ are both incident to the face where $\Gamma(\mu)$ has been inserted. Finally, insert the outer edges of μ. This can always be done so that no outer edge of μ in G_1 (resp. in G_2) crosses an edge of $G_1(\mu)$ (resp. of $G_2(\mu)$), by the assumption that the outer edges of μ can be drawn towards the outer face, and so that no outer edge of μ in G_1 (resp. in G_2) crosses an edge of the graph obtained from G_1 be removing $G_1(\mu)$, except for its poles (resp. of the graph obtained from G_2 be removing $G_2(\mu)$, except for its poles), since the drawing of such outer edges in Γ used to exist and $L_1^a(\Gamma,\mu) = L_1^a(\Gamma(\mu),\mu)$, $L_1^b(\Gamma,\mu) = L_1^b(\Gamma(\mu),\mu)$, $L_2^a(\Gamma,\mu) = L_2^a(\Gamma(\mu),\mu)$, and $L_2^b(\Gamma,\mu) = L_2^b(\Gamma(\mu),\mu)$. □

Finally, the following claim asserts that the algorithm computes a SEFE of $G_1(\mu)$ and $G_2(\mu)$, if it exists, in which the structure of the outer face of $G_{1\cap 2}(\mu)$ is the one that allows for an extension into a SEFE of G_1 and G_2.

Claim 3 *If, for any node $\mu \in \mathcal{T}$ that is not an S-node, a* SEFE *of $G_1(\mu)$ and $G_2(\mu)$ exists in which the outer edges of μ can be drawn towards the outer face, then the algorithm computes one.*

Proof: Suppose, for a contradiction, that there exists a node that is not an S-node and for which the statement does not hold and consider a node $\mu \in \mathcal{T}$ that is not an

S-node, for which the statement does not hold, and such that all the descendants of μ in $\mathcal{T}$ satisfy the statement.

The pertinent graph of each Q-node has exactly one embedding and the statement trivially holds for such nodes. Hence μ can not be a Q-node.

If μ is a P-node, then by construction the algorithm chooses the embedding of $skel(\mu)$ and flips the visible nodes of μ in such a way that the end-vertices of the outer edges of μ are incident to the outer face of $G_{1\cap 2}(\mu)$. Hence, if μ is a P-node, it suffices to show that the algorithm constructs a SEFE $\Gamma(\mu)$ of $G_1(\mu)$ and $G_2(\mu)$, if a SEFE of $G_1(\mu)$ and $G_2(\mu)$ exists in which the outer edges of μ can be drawn towards the outer face.

Consider the graph O which has a vertex for each virtual edge in $skel(\mu)$, and which has an edge between two vertices if the corresponding nodes share an outer edge. If O is not a simple cycle or a collection of paths, then, by Lemma 1, a SEFE of G_1 and G_2 does not exist, and there is nothing to prove. If O is a simple cycle or a collection of paths, denote by $O_1, O_2 \dots, O_k$ the connected components of O. For $i = 1, \dots, k$, denote by $G_1(O_i)$, $G_2(O_i)$, and $G_{1\cap 2}(O_i)$ the subgraph of G_1, of G_2, and of $G_{1\cap 2}$, respectively, induced by the vertices belonging to nodes of $\mathcal{T}$ in O_i. Notice that, in order to prove that the algorithm computes a SEFE of $G_1(\mu)$ and $G_2(\mu)$ if one exists in which the outer edges of μ can be drawn towards the outer face, graphs $G_1(O_i)$, $G_2(O_i)$, and $G_{1\cap 2}(O_i)$ can be treated separately from graphs $G_1(O_j)$, $G_2(O_j)$, and $G_{1\cap 2}(O_j)$, for any $j \neq i$. Namely, there exists no exclusive edge of G_1 or of G_2 connecting a vertex different from $u(\mu)$ and $v(\mu)$ in $G_{1\cap 2}(O_i)$ with a vertex different from $u(\mu)$ and $v(\mu)$ in $G_{1\cap 2}(O_j)$, as otherwise O_i and O_j would not be distinct. Hence, if all the pairs $G_1(O_i)$ and $G_2(O_i)$, with $1 \leq i \leq k$, admit a SEFE, a SEFE of $G_1(\mu)$ and $G_2(\mu)$ can be computed by placing all such SEFEs one beside the other and by continuously deforming the drawing of the edges incident to $u(\mu)$ and $v(\mu)$ till $u(\mu)$ and $v(\mu)$ are placed at the same point in all such SEFEs.

Now consider a component O_i composed of nodes $\nu_x, \nu_{x+1}, \dots, \nu_y$ (possibly one of such nodes represents the rest of the graph). By induction and by Claim 2, for each visible node τ of μ, the algorithm computes a SEFE of $G_1(\tau)$ and $G_2(\tau)$ in which the outer edges of τ can be drawn towards the outer face. Moreover, observe that, by Lemma 1, the order of nodes $\nu_x, \nu_{x+1}, \dots, \nu_y$ induced by any SEFE of $G_1(O_i)$ and $G_2(O_i)$ is fixed, up to a reversal of such an ordering. Hence, what we have to prove is that the flips of the visible nodes of μ and the edge embeddings that are decided by the algorithm do not alter the possibility of finding a SEFE of $G_1(O_i)$ and $G_2(O_i)$.

Observe that some choices of the algorithm are forced. Namely, if a visible node τ_1 has an outer edge e incident to it in a vertex $u(e)$, if a visible node $\tau_2 \neq \tau_1$ has e as an outer edge incident to it in a vertex $v(e)$, and if τ_1 and τ_2 are not children of the same S-node ν_i, then the flips of τ_1 and τ_2 are the only ones that bring $u(e)$ and $v(e)$ to be incident to the face shared by τ_1 and τ_2. Consequently, the embedding of the outer edges of τ_1 and τ_2 is also forced to be in a certain face. In turn, the constraint on such edges forces the flips of some components, and such flips again force the embedding of the outer edges of the components, and so on. Hence, while the algorithm flips such components and embeds such edges, it makes no arbitrary choices.

After such forced choices are all done (in the algorithm's description when $E_f = \emptyset$ and $T_f = \emptyset$), either a SEFE of $G_1(\mu)$ and $G_2(\mu)$ has been computed, or some visible nodes still have to be flipped and some edges still have to embedded. However, observe that: (i) no edge to be embedded potentially crosses with an already embedded edge; (ii) no edge to be embedded is the outer edge of an already flipped visible node of μ; (iii) no visible node of μ that still has to be flipped has an outer edge that has been already embedded. Consider any edge e still to be embedded. Embed it in any planar way. Once again, such a choice sequentially forces the flip of some components and the embedding of some edges. Denote by $C(e)$ the set of components that are flipped after choosing the embedding of e and by $E(e)$ the set of edges that are embedded after choosing the embedding of e. First, observe that all such edges belong to graphs $G_1(\nu_i)$ or $G_2(\nu_i)$, for a certain $1 \leq i \leq k$, as otherwise an edge between two nodes ν_i and ν_j would exist in $E(e)$ and such an edge would have been embedded at the first step of the algorithm. Hence, ν_i is an S-node, and the edges in $E(e)$ are outer edges for the children of ν_i. If the algorithm fails with the choice done for e, then consider the opposite choice for e. Then, the algorithm would make opposite choices for every element of $C(e)$ or of $E(e)$ and it would still fail, thus contradicting the fact that $G_1(\mu)$ and $G_2(\mu)$ have a SEFE.

The case of the R-nodes is similar to and simpler than the one of the P-nodes. □

Claims 1–3 prove the second implication of the lemma. Namely, when μ is the child of the root of $\mathcal{T}$, the algorithm computes a SEFE $\Gamma(\mu)$ of $G_1(\mu)$ and $G_2(\mu)$, by Claim 3. Observe that μ is not an S-node, by construction of $\mathcal{T}$. Also, observe that μ has no outer edge. By Claim 2, $\Gamma(\mu)$ can be extended into a SEFE Γ of G_1 and G_2, if such a SEFE exists. To this end, however, it is sufficient to draw the edge that is the root of $\mathcal{T}$. □

We get the following.

Theorem 1 *Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two planar graphs on the same set of n vertices such that the intersection graph $G_{1\cap 2}$ is biconnected. Then, it is possible to test whether G_1 and G_2 admit a* SEFE *in $O(n^3)$ time.*

Proof: We show how to implement the algorithm described in this section in $O(n^3)$ time. The correctness of the algorithm has already been proved in Lemma 3.

First, we need to compute some labels for the edges of G_1 and G_2 and for the nodes of $\mathcal{T}$. Each edge of G_1 and each edge of G_2 is equipped with a label indicating whether the edge belongs to $G_{1\cap 2}$, to $G_{1\setminus 2}$, or to $G_{2\setminus 1}$. All such labels can be set up in total $O(n)$ time by visiting G_1 and G_2. Thus, graphs $G_{1\cap 2}$, $G_{1\setminus 2}$, and $G_{2\setminus 1}$ can be constructed in $O(n)$ time. The SPQR-tree $\mathcal{T}$ of $G_{1\cap 2}$ can be constructed in $O(n)$ time [4, 5, 14]. Further, for each exclusive edge e of G_1 and G_2, we equip each node μ with a label $\ell(\mu, e)$ indicating whether e is an outer edge of μ, or an internal edge of μ, or an intra-pole edge of μ, or none of the previous ones.

Claim 4 *Setting up label $\ell(\mu, e)$, for all the pairs (μ, e), can be done in $O(n^2)$ time.*

Proof: Let $e = (u, v)$ be an exclusive edge of G_1 or of G_2 The nodes of $\mathcal{T}$ that contain u and the nodes of $\mathcal{T}$ that contain v form subtrees $\mathcal{T}_u$ and $\mathcal{T}_v$ of $\mathcal{T}$, respectively. As there exist nodes of $\mathcal{T}$ that contain both u and v, the union of $\mathcal{T}_u$ and $\mathcal{T}_v$ is a subtree of $\mathcal{T}$ that we denote by $\mathcal{T}_e$. Tree $\mathcal{T}_u$ can be computed in $O(n)$ time, by means of a bottom-up traversal of $\mathcal{T}$ that starts by marking the Q-nodes that contain u and continues by marking a node of $\mathcal{T}$ such that one of its children is marked. Trees $\mathcal{T}_v$ and $\mathcal{T}_e$ can be analogously computed in $O(n)$ time.

By traversing $\mathcal{T}_e$, $\mathcal{T}_u$, and $\mathcal{T}_v$, it is possible to assign a label $h(\mu, e)$ to each node $\mu \in \mathcal{T}$, for each exclusive edge e of G_1 or of G_2, indicating whether e has both the end-vertices in μ, one of its end-vertices in μ, or none of its end-vertices in μ. Namely, all the nodes of $\mathcal{T}_e$ contain both the end-vertices of e, all the nodes of $\mathcal{T}_u$ and $\mathcal{T}_v$ not in $\mathcal{T}_e$ contain one of the end-vertices of e, and all the other nodes of $\mathcal{T}$ contain none of the end-vertices of e. Hence, for each exclusive edge e of G_1 or of G_2, all the nodes $\mu \in \mathcal{T}$ can be labeled in $O(n)$ time and hence, summing over all the exclusive edges of G_1 or of G_2, labels $h(\mu, e)$ can be computed in total $O(n^2)$ time.

For an exclusive edge $e = (u, v)$ of G_1 or of G_2, we define its *attachment* $u_\mu(e)$ (resp. $v_\mu(e)$) in μ, as u (resp. v) if u (resp. v) is a vertex of $skel(\mu)$ and as the virtual edge of $skel(\mu)$ that contains u (resp. v) otherwise. For each exclusive edge $e = (u, v)$ of G_1 or of G_2, we can easily compute $u_\mu(e)$ and $v_\mu(e)$ in $O(n)$ time over all the nodes μ by a bottom-up traversal of $\mathcal{T}_e$. Hence, the attachments $u_\mu(e)$ and $v_\mu(e)$, over all the pairs (μ, e), can be computed in total $O(n^2)$ time.

Now, consider an exclusive edge e of G_1 or of G_2. Checking whether e is an outer edge of a given node $\mu \in \mathcal{T}_e$ can be done in constant time, as we only need to check that either $u_\mu(e)$ or $v_\mu(e)$ is the virtual edge of $skel(\mu)$ that represents the rest of the graph and that the one of $u_\mu(e)$ and $v_\mu(e)$ that is not the virtual edge of $skel(\mu)$ is not a pole of μ. Hence, all the outer edges for all the nodes of $\mathcal{T}$ can be recognized in total $O(n^2)$ time. Similarly, checking whether e is an intra-pole edge of a given node $\mu \in \mathcal{T}_e$ can be done in constant time, hence all the intra-pole edges for all the nodes of $\mathcal{T}$ can be recognized in total $O(n^2)$ time. For recognizing the only node of $\mathcal{T}_e$ for which e is an internal edge, if any, we perform a bottom-up traversal of $\mathcal{T}_e$ that works as follows. While the node of μ for which e is internal has not been found, a node μ is visited and if the node contains both the end-vertices of e, as stored in $h(\mu, e)$, none of its children contains both the end-vertices of e, and e is not the intra-pole edge of μ, then μ is the wanted node. Hence, all the internal edges for all the nodes of $\mathcal{T}$ can be recognized in total $O(n^2)$ time. □

We are now ready to describe how to make the algorithm presented in this section run in $O(n^3)$ time. Determining an embedding for the skeletons of the nodes of $\mathcal{T}$ can be done quickly, as shown in the following.

Claim 5 *Choosing an embedding for $skel(\mu)$, for all the nodes $\mu \in \mathcal{T}$, can be done in $O(n^2)$ time.*

Proof: We actually prove a stronger statement, namely that choosing an embedding for $skel(\mu)$ can be done in $O(n)$ time for each node $\mu \in \mathcal{T}$.

If μ is a Q-, S-, or an R-node, then $skel(\mu)$ has only one embedding up to a flip, hence an embedding for $skel(\mu)$ can be fixed in $O(n)$ time. If μ is a P-node, we first compute graph O. In order to do this, we consider all the internal and outer edges of μ. For each such an edge e, its attachments $u_\mu(e)$ and $v_\mu(e)$ are virtual edges of $skel(\mu)$. We then add an edge connecting such virtual edges in O. When we are adding an edge to O, we check if it is already present in O and in case we discard it. Also, when we are adding an edge to O, we check if an end-vertex of O already has two distinct incident edges and in case we reject the instance. To accomplish both checks in constant time, it is sufficient to store the edges incident to each vertex of O when they are added to O. If we did not reject earlier, then O has maximum degree 2 and we can check if O is a cycle or a collection of paths in $O(n)$ time. In this case an ordering of the virtual edges of μ around $u(\mu)$ such that two vertices adjacent in O are consecutive in the ordering can be easily constructed in $O(n)$ time. □

Next, we show how to implement the Embedding-Flipping step in $O(n^2)$ time for each node $\mu \in \mathcal{T}$ that is not an S-node. For this sake, we construct graphs F_k^1 and F_k^2 for each face f_k of the computed embedding of $skel(\mu)$. We construct the vertex sets of all the graphs F_k^1 and F_k^2 for a certain node μ as follows. Consider an exclusive edge e of G_1 (resp. of G_2). If e is an intra-pole edge of μ, then add it to every graph F_k^1 (resp. F_k^2). If e is an internal edge or an outer edge of μ, then consider the virtual edges of $skel(\mu)$ that contain $u_\mu(e)$ and $v_\mu(e)$; such virtual edges are incident to the same face f_k of the embedding of $skel(\mu)$; then add e to graph F_k^1 (resp. F_k^2). If e is an internal edge of an S-node child of μ, then consider the virtual edge of $skel(\mu)$ that represents such an S-node; such a virtual edge is incident to two faces f_k and $f_{k'}$ of the embedding of $skel(\mu)$; then add e to graphs F_k^1 and $F_{k'}^1$ (resp. F_k^2 and $F_{k'}^2$). As for each edge such computations can easily be performed in total $O(n)$ time, the vertex sets of all the graphs F_k^1 and F_k^2, for a certain node μ in $\mathcal{T}$, can be computed in $O(n^2)$ time. Next, we compute the edges of graphs F_k^1 and F_k^2. For each face f_k of the embedding of $skel(\mu)$, order the vertices and the virtual edges incident to f_k as they appear in the two paths from $u(\mu)$ and $v(\mu)$ delimiting f_k; then two vertices in F_k^1 (resp. in F_k^2) are adjacent if the corresponding edges (u, v) and (u', v') are such that $u \prec u' \prec v \prec v'$ when all such vertices are in the same virtual edge of $skel(\mu)$, or such that $u \prec u' \prec v$ when vertices u, v, and u' are in the same virtual edge of $skel(\mu)$ and v' is in a different one, or such that $u \prec u'$ and $v' \prec v$ when vertices u and u' are in the same virtual edge of $skel(\mu)$ and v and v' are in a different one. After an $O(n)$-time processing to set up such orders, each pair of edges can be handled in $O(1)$ time. Since there are $O(n^2)$ pairs of edges, graphs F_k^1 and F_k^2 can be constructed in total $O(n^2)$ time for each node $\mu \in \mathcal{T}$. Once graphs F_k^1 and F_k^2 have been constructed, the Embedding-Flipping step is easily performed in $O(n^2)$ time for each node $\mu \in \mathcal{T}$. Hence, the Embedding-Flipping steps for all the nodes in $\mathcal{T}$ require $O(n^3)$ time.

The final planarity testing of each internal face f of $G_{1\cap 2}(\mu)$ corresponding to a face of $skel(\mu)$, augmented with the edges of $G_{1\setminus 2}$ or of $G_{2\setminus 1}$, is performed in linear time in the number of vertices incident to f. □

4 The Intersection Graph is a Tree

In this section we show that the SEFE problem, when the intersection graph is a tree, is equivalent to a 2-page book embedding problem defined in the following.

Let G be a graph, let (E_1, E_2) be a partition of its edge set, and let T be a rooted tree whose leaves are the vertices of G. Problem PARTITIONED T-COHERENT 2-PAGE BOOK EMBEDDING with input (G, E_1, E_2, T) asks: Does a 2-page book embedding of G exist in which the edges of E_1 lie in one page, the edges of E_2 lie in the other page, and, for every internal vertex $t \in T$, the vertices of G in the subtree of T rooted at t appear consecutively in the vertex ordering of G defined in the book embedding?

We now show how to transform an instance $G_1 = (V, E_1), G_2 = (V, E_2)$ of SEFE in which $G_{1\cap 2}$ is a tree into an instance of PARTITIONED T-COHERENT 2-PAGE BOOK EMBEDDING. Such a transformation consists of two steps.

In the first step, we transform instance G_1, G_2 of SEFE into an equivalent instance G'_1, G'_2 of SEFE such that $G'_{1\cap 2}$ is a tree and all the exclusive edges of G'_1 and of G'_2 are incident only to leaves of $G'_{1\cap 2}$. To this end, we modify every edge $(u, v) \in G_{1\setminus 2}$ such that u is not a leaf of $G_{1\cap 2}$ as follows. We subdivide edge (u, v) with a new vertex u'; we add edge (u, u') to E_2, so that u' is a leaf in the intersection graph of the two modified graphs. Symmetrically, we subdivide every edge $(u, v) \in G_{2\setminus 1}$ such that u is not a leaf of $G_{1\cap 2}$ with a new vertex u' and we add edge (u, u') to E_1, so that u' is a leaf in the intersection graph of the two modified graphs. Note that the exclusive edges of G_1 and G_2 that are incident to two non-leaf vertices are subdivided twice. Denote by G'_1 and by G'_2 the resulting graphs. We have the following:

Lemma 4 *G_1, G_2 is a positive instance of* SEFE *if and only if G'_1, G'_2 is a positive instance of* SEFE. *Further, $G'_{1\cap 2}$ is a tree and all the exclusive edges of G'_1 and of G'_2 are incident only to leaves of $G'_{1\cap 2}$. Moreover, $G'_{1\cap 2}$ has $O(n)$ vertices.*

Proof: $G_{1\cap 2}$ is a tree, by assumption. When an exclusive edge (u, v) in G_1 (resp. in G_2) such that u is not a leaf of $G_{1\cap 2}$ is subdivided with a vertex u' and edge (u, u') is added to E_2 (resp. to E_1), an edge is inserted into $G_{1\cap 2}$ connecting an internal vertex of $G_{1\cap 2}$ with a new leaf of $G_{1\cap 2}$, namely u'. Hence, $G_{1\cap 2}$ remains a tree after such a modification and thus $G'_{1\cap 2}$ is a tree. When an exclusive edge (u, v) in G_1 (resp. in G_2) such that u is not a leaf of $G_{1\cap 2}$ is subdivided with a vertex u' and edge (u, u') is added to E_2 (resp. to E_1), the number of incidences between exclusive edges and internal vertices of $G_{1\cap 2}$ decreases by one. Hence, after all such modifications have been performed, all the exclusive edges are incident only to leaves of $G'_{1\cap 2}$. Each exclusive edge is subdivided at most twice. Since the number of edges of $G_{1\setminus 2}$ and $G_{2\setminus 1}$ is $O(n)$, then $G'_{1\cap 2}$ has $O(n)$ vertices. We now prove that G_1, G_2 is a positive instance of SEFE if and only if G'_1, G'_2 is a positive instance of SEFE.

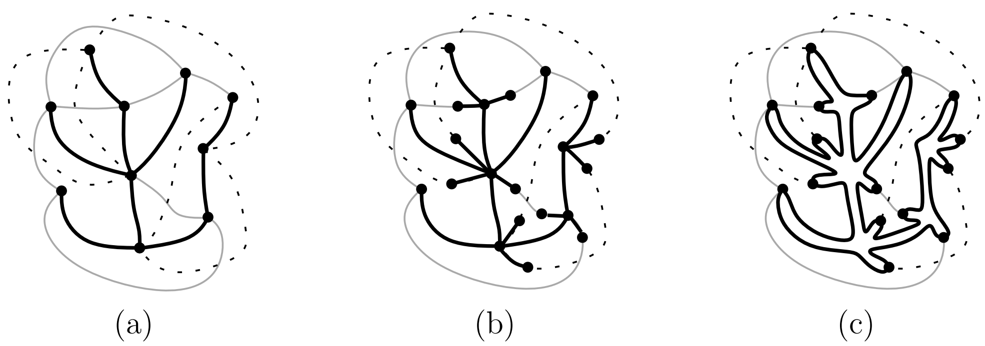

Figure 3: (a) A SEFE Γ of G_1, G_2. (b) A SEFE Γ' of G'_1, G'_2. The edges of $G_{1\cap 2}$ and of $G'_{1\cap 2}$ are thick lines, the edges of $G_{1\setminus 2}$ and of $G'_{1\setminus 2}$ are drawn gray, and the edges of $G_{2\setminus 1}$ and of $G'_{2\setminus 1}$ are black dotted lines. (c) Euler Tour $\mathcal{E}$ of $G'_{1\cap 2}$ and exclusive edges.

First, suppose that a SEFE Γ of G_1, G_2 exists. Modify Γ to obtain a SEFE Γ' of G'_1, G'_2 as follows (see Figs. 3(a) and 3(b)). When an exclusive edge (u, v) in G_1 (resp. in G_2) such that u is not a leaf of $G_{1\cap 2}$ is subdivided with a vertex u' and edge (u, u') is added to E_2 (resp. to E_1), insert u' in Γ along edge (u, v) arbitrarily close to u. Since the drawing of G_1 in Γ is not modified and since the drawing of G_2 in Γ is modified by inserting an arbitrarily small edge incident to a vertex, the resulting drawing is a SEFE of the current graphs and hence Γ' is a SEFE of G'_1, G'_2. Second, suppose that a SEFE Γ' of G'_1, G'_2 exists. A SEFE Γ of G_1, G_2 can be obtained by drawing each edge (u, v) of G_1 (resp. of G_2) exactly as in Γ'. Observe that (u, v) is subdivided never, once, or twice in G'_1 (resp. in G'_2); then, its drawing in Γ is composed of the concatenation of the one, two, or three curves representing the parts of (u,v) in Γ'. That no two edges of G_1 (resp. of G_2) intersect in the resulting drawing Γ directly descends from the fact that no two edges of G'_1 (resp. of G'_2) intersect in Γ'. □

In the second step, we transform an instance G_1, G_2 of SEFE such that $G_{1\cap 2}$ is a tree and all the exclusive edges of G_1 and of G_2 are incident only to leaves of $G_{1\cap 2}$ into an equivalent instance of PARTITIONED T-COHERENT 2-PAGE BOOK EMBEDDING.

The input of PARTITIONED T-COHERENT 2-PAGE BOOK EMBEDDING consists of the graph G composed of all the vertices which are leaves of $G_{1\cap 2}$, of all the exclusive edges $E_{1\setminus 2}$ of $G_{1\setminus 2}$, and of all the exclusive edges $E_{2\setminus 1}$ of $G_{2\setminus 1}$. The partition of the edges of G is $(E_{1\setminus 2}, E_{2\setminus 1})$. Finally, tree T is $G_{1\cap 2}$. We have the following:

Lemma 5 *G_1, G_2 is a positive instance of* SEFE *if and only if $(G, E_{1\setminus 2}, E_{2\setminus 1}, T)$ is a positive instance of* PARTITIONED T-COHERENT 2-PAGE BOOK EMBEDDING.

Proof: Suppose that $(G, E_{1\setminus 2}, E_{2\setminus 1}, T)$ is a positive instance of PARTITIONED T-COHERENT 2-PAGE BOOK EMBEDDING. See Fig. 4. An ordering of the vertices of G along a line ℓ exists such that the edges in $E_{1\setminus 2}$ are drawn on one side ℓ^+ of ℓ, the edges in $E_{2\setminus 1}$ are drawn on the other side ℓ^- of ℓ, no two edges in $E_{1\setminus 2}$ cross, and no two edges in $E_{2\setminus 1}$ cross. Move all the edges in $E_{2\setminus 1}$ to ℓ^+. Since such edges do not cross in ℓ^- and since the ordering of the vertices of G is not modified, the edges in $E_{2\setminus 1}$ still do not cross. Finally, construct a planar drawing of $G_{1\cap 2}$ in ℓ^-. This can always be done since, for each internal vertex t of $G_{1\cap 2}$, the vertices in the subtree of $G_{1\cap 2}$ rooted at t appear consecutively on ℓ. The resulting drawing is hence a SEFE of G_1, G_2.

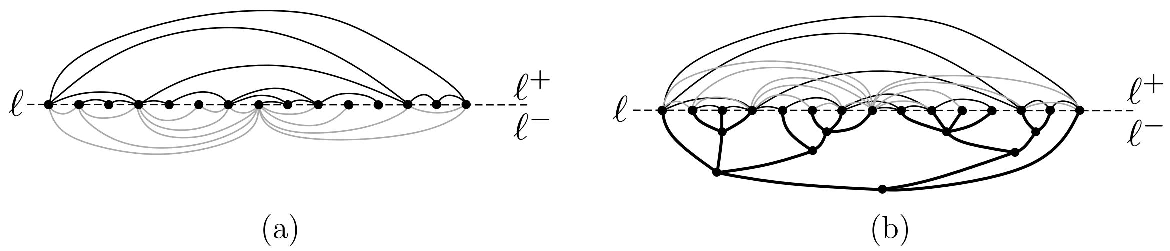

Figure 4: (a) A Partitioned T-coherent 2-page book embedding of $(G, E_{1\setminus 2}, E_{2\setminus 1}, T)$. The edges in $E_{1\setminus 2}$ are thin black lines, the edges in $E_{2\setminus 1}$ are thin gray lines. (b) The SEFE of G_1, G_2 obtained from the book embedding of $(G, E_{1\setminus 2}, E_{2\setminus 1}, T)$.

Suppose that G_1, G_2 is a positive instance of SEFE. Consider any SEFE Γ of G_1, G_2 and consider an Euler Tour $\mathcal{E}$ of $G_{1\cap 2}$. Construct a planar drawing of $\mathcal{E}$ in Γ as follows. Each edge of $\mathcal{E}$ is drawn arbitrarily close to the corresponding edge in $G_{1\cap 2}$. Each end-vertex t of an edge of $\mathcal{E}$ that is a leaf in $G_{1\cap 2}$ is drawn at the same point where it is drawn in Γ. Each end-vertex t of an edge of $\mathcal{E}$ that is not a leaf in $G_{1\cap 2}$ and that has two adjacent edges (t, t_1) and (t, t_2) in $\mathcal{E}$ (observe that $t_1 \neq t_2$ as t is an internal vertex of $G_{1\cap 2}$) is drawn arbitrarily close to the point where t is drawn in Γ, in the region "between" edges (t, t_1) and (t, t_2). Clearly, the resulting drawing of $\mathcal{E}$ is planar (see Figs. 3(b) and 3(c)). Further, all the leaf vertices of $G_{1\cap 2}$ are drawn at the same point in Γ and in the drawing of $\mathcal{E}$. Moreover, all the exclusive edges of $G_{1\setminus 2}$ and all the exclusive edges of $G_{2\setminus 1}$ lie entirely outside $\mathcal{E}$, except for their end-vertices. Remove all the internal vertices and all the edges of $G_{1\cap 2}$ from the drawing. Move all the edges of $G_{2\setminus 1}$ inside $\mathcal{E}$. The resulting drawing is a Partitioned T-coherent 2-page book embedding of $(G, E_{1\setminus 2}, E_{2\setminus 1}, T)$. Namely, all the edges in $E_{1\setminus 2}$ are on one side of $\mathcal{E}$ and all the edges in $E_{2\setminus 1}$ are on the other side of $\mathcal{E}$. No two edges in $E_{1\setminus 2}$ cross as they do not cross in Γ. No two edges in $E_{2\setminus 1}$ cross as they do not cross in Γ. Finally, all the leaf vertices in a subtree of $G_{1\cap 2}$ rooted at an internal vertex t of $G_{1\cap 2}$ appear consecutively in $\mathcal{E}$, as the drawing of $G_{1\cap 2}$ in Γ is planar. □

Given an instance (G, E_1, E_2, T) of Partitioned T-coherent 2-page book embedding, it is possible to construct an equivalent instance of Sefe as follows. Let G_1 be the graph whose vertex set is composed of the vertices of G and of the internal vertices of T, and whose edge set is composed of the edges of E_1 and of the edges of T. Analogously, let G_2 be the graph whose vertex set is composed of the vertices of G and of the internal vertices of T, and whose edge set is composed of the edges of E_2 and of the edges of T. Analogous to Lemma 5, we can prove the following lemma.

Lemma 6 (G, E_1, E_2, T) *is a positive instance of* Partitioned T-coherent 2-page book embedding *if and only if* G_1, G_2 *is a positive instance of* Sefe.

Since both reductions can easily be performed in linear time we obtain the following.

Theorem 2 Partitioned T-coherent 2-page book embedding *and* Sefe *have the same time complexity.*

The problem Partitioned T-coherent 2-page book embedding has been recently studied by Hong and Nagamochi [15] when T is a star. That is, the graph has the edges partitioned into two pages as part of the input, but there is no constraint on the order of the vertices in the required book embedding. In such a case, Hong and Nagamochi proved that the problem is $O(n)$-time solvable [15]. While their motivation was a connection to the c-planarity problem, Lemmata 4 and 5 together with Hong and Nagamochi's result imply that deciding whether a Sefe exists for two graphs whose intersection graph is a star is a linear-time solvable problem.

Theorem 3 *The* Sefe *problem is solvable in linear time when the intersection graph is a star.*

5 Conclusions

In this paper we have shown new results on the time complexity of the problem of deciding whether two planar graphs admit a Sefe.

First, we have shown that the Sefe problem can be solved in cubic time if the intersection graph $G_{1\cap 2}$ of the input graphs G_1 and G_2 is biconnected. We believe that a refined implementation of our approach could reduce such a time bound to quadratic. More in general, with similar techniques we can solve in polynomial time the Sefe problem if $G_{1\cap 2}$ consists of one biconnected component plus a set of isolated vertices. Also, the following generalization of the Sefe problem with $G_{1\cap 2}$ biconnected seems worth to be tackled: What is the time complexity of computing a Sefe when $G_{1\cap 2}$ is *edge-biconnected*?

Second, we have shown that when $G_{1\cap 2}$ is a tree the Sefe problem can be equivalently stated as a 2-page book embedding problem with edges assigned to the pages and with hierarchical constraints. Hence, pursuing an $\mathcal{NP}$-hardness proof for such a book embedding problem is a possible direction for trying to prove the $\mathcal{NP}$-hardness for the Sefe problem.

Acknowledgments

Thanks to Seok-Hee Hong for presenting the results in [15] to the authors.

References

[1] P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. In *Symposium on Discrete Algorithms (SODA '10)*, pages 202–221, 2010.

[2] P. Angelini, M. Geyer, M. Kaufmann, and D. Neuwirth. On a tree and a path with no geometric simultaneous embedding. *CoRR*, 2010.

[3] P. Braß, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous planar graph embeddings. *Comput. Geom.*, 36(2):117–130, 2007.

[4] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996.

[5] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996.

[6] E. Di Giacomo and G. Liotta. Simultaneous embedding of outerplanar graphs, paths, and cycles. *Int. J. Comput. Geometry Appl.*, 17(2):139–160, 2007.

[7] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. *J. Graph Algorithms Appl.*, 9(3):347–364, 2005.

[8] A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous geometric graph embeddings. In S. H. Hong, T. Nishizeki, and W. Quan, editors, *Graph Drawing (GD '07)*, volume 4875 of *LNCS*, pages 280–290, 2007.

[9] J. J. Fowler, C. Gutwenger, M. Jünger, P. Mutzel, and M. Schulz. An SPQR-tree approach to decide special cases of simultaneous embedding with fixed edges. In I. G. Tollis and M. Patrignani, editors, *Graph Drawing (GD '08)*, volume 5417 of *LNCS*, pages 157–168, 2008.

[10] J. J. Fowler, M. Jünger, S. G. Kobourov, and M. Schulz. Characterizations of restricted pairs of planar graphs allowing simultaneous embedding with fixed edges. In H. Broersma, T. Erlebach, T. Friedetzky, and D. Paulusma, editors, *Graph-Theoretic Concepts in Computer Science (WG '08)*, volume 5344 of *LNCS*, pages 146–158, 2008.

[11] F. Frati. Embedding graphs simultaneously with fixed edges. In M. Kaufmann and D. Wagner, editors, *Graph Drawing (GD '06)*, volume 4372 of *LNCS*, pages 108–113, 2006.

[12] E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous graph embeddings with fixed edges. In F. V. Fomin, editor, *Graph-Theoretic Concepts in Computer Science (WG '06)*, volume 4271 of *LNCS*, pages 325–335, 2006.

[13] M. Geyer, M. Kaufmann, and I. Vrt'o. Two trees which are self-intersecting when drawn simultaneously. *Discrete Mathematics*, 307:1909–1916, 2009.

[14] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing (GD '00)*, volume 1984 of *LNCS*, pages 77–90, 2001.

[15] S. H. Hong and H. Nagamochi. Two-page book embedding and clustered graph planarity. Tech. Report 2009-004, Department of Applied Mathematics & Physics, Kyoto University, 2009.

[16] M. Jünger and M. Schulz. Intersection graphs in simultaneous embedding with fixed edges. *J. Graph Alg. & Appl.*, 13(2):205–218, 2009.